Lightning Talk Abstracts

ICER 2015

We are pleased to again include a collection of lightning talks as part of the ICER conference program. These talks intend to further expand the ICER community and spark discussion among conference participants. The goal is for these talks to provide a venue in support of both new ideas and newcomers to our community. New to ICER, some of the talks will have accompanying posters (in Poster Session 2) which are designed to help continue the conversation.

Like the main ICER program, the talks cover a wide range of computing education related work, and the presenters are from a variety of different institutions around the world. Short abstracts for each of the talks to be presented at this year's conference can be found on the pages to follow.

Leo Porter
2015 Lightning Talks and Posters Chair
UC San Diego

Lightning Talks

1. *Power and Epistemology in the CS Classroom*
   Donald Chinn and Josh Tenenberg
   University of Washington Tacoma, USA

2. *Educating the Next Generation of Great Software Engineers*
   Leigh Ann DeLyser
   NYC Foundation for CS Education, USA

3. *The MoveLab: Supporting Diversity through Self-Conceptions*
   Kayla DesPortes
   Georgia Institute of Technology, USA

4. *Targeting Intervention Using A Grammar For The Triage of Questions in an On-line Discussion Forum*
   Nickolas Falkner and Claudia Szabo
   The University of Adelaide, Australia

5. *Pedagogical Content Knowledge of Veteran Teachers New to CS*
   Aleata Hubbard
   WestEd

6. *Computer Science is "Hard": Looking at the Gender Gap Between Two Computing Programs*
   Brittany Ann Kos
   ATLAS Institute and University of Colorado Boulder, USA

7. *Middle School Students Learn Programming Using Wearable Technologies*
   Victoria Lentfer
   University of Nebraska Omaha, USA

8. *Storyteller: Telling Stories by Reflecting on Code Evolution*
   Mark Mahoney
   Carthage College, USA

9. *Gender Differences in High School Students' Decisions to Study Computer Science and Related Fields*
   Jason Ravitz
   Google, USA

10. *How to Train Your Robot: Computing Introduction for 8-10 Year Olds*
    Chad Williams, Emtethal Alafghani, Antony Daley Jr., Kevin Gregory, and Marianella Rydzewski
    Central Connecticut State University, USA

Power and Epistemology in the CS Classroom

Donald Chinn, Josh Tenenberg
University of Washington Tacoma
{dchinn, jtenenbg}@u.washington.edu

Abstract:
Our research focuses on the relationship between the epistemic and social orders of the computer science classroom. More specifically, what beliefs do instructor and students have about the nature of disciplinary knowledge and their respective roles in the classroom? How are these beliefs described? How are these beliefs enacted in the classroom?

These questions are related to issues of power, authority, and legitimacy. Who is allowed to speak in class and when? How is knowledge created in the classroom (e.g. "sage on the stage" vs. "guide on the side")? What constitutes the authoritativeness of actors and of knowledge? How are students expected to interact with each other? How is it that students have confidence that the instructor knows the material?

This work attempts to empirically probe these issues through a case study in a way that can inform instructors' intentional design of CS classroom environments. Our data sources are from a data structures course taught at a liberal arts university in the United States. The data includes video from classroom and lab sessions throughout the academic term, as well as pre- and post-course interviews with the instructor and a dozen students.

Our work is influenced by the work of William Perry, who studied the intellectual development of college students. Perry observed that, epistemologically speaking, students tend to go from a black and white view of knowledge (what he called Dualism) to that of a more nuanced relativistic view of knowledge. Our work is also influenced by the work within the CS Education community to understand the CS classroom through the lens of power (e.g. classroom climate work of Lecia Barker et al.).

Educating the Next Generation of Great Software Engineers

Leigh Ann DeLyser
NYC Foundation for CS Education
leighann@csnyc.org

Abstract:
Career and technical education (CTE) programs at the K-12 school level provide students with the opportunity to study standard academic courses, as well as specialize in career themed programs of study. In New York City, the Academy for Software Engineering (AFSE) is a CTE school focused on software engineering. As the first of its kind, AFSE has had to construct a program that encompasses academic coursework, rigorous standards with an end-of-program technical assessment, and work based learning, where students engage with industry professionals.

In the fall of 2015, AFSE will submit its program for NY State approval. Over the past three years the program has been developed, classroom tested, and refined to meet the needs of NYC public school students and, at the same time, expectations of partner companies who hire software engineers. Companies such as Google, Morgan Stanley, JP Morgan, Facebook, and others have all participated in the design and evaluation of the program.

To connect the program of study to the ICER community, the presentation will highlight similarities between the three parts of the AFSE program and previous works focusing on the necessary content in software engineering education [1], the tasks performed by novice software engineers [2], and desired qualities of expert software engineers [3]. It is the hope of the author that conversation with the ICER community can influence final program adjustments before the CTE program is submitted for state approval.

[1] Tucker, A. (2003). A Model Curriculum for K--12 Computer Science: Final Report of the ACM K--12 Task Force Curriculum Committee.

[2] Begel, A., & Simon, B. (2008). Novice software developers, all over again. In *Proceedings of the Fourth international Workshop on Computing Education Research* (pp. 3-14). ACM.

[3] Li, P., Ko, A.J., and Zhu, J. (2015) What Makes a Great Software Engineer?, International Conference on Software Engineering. IEEE

The MoveLab: Supporting Diversity through Self-Conceptions

Kayla DesPortes
Georgia Institute of Technology
ksdesportes@gatech.edu

Abstract:
Individuals within underrepresented populations in computing often do not perceive the identity of a *computer scientist* as aligning with their interests or value system. In many cases this leads to rejection of opportunities to participate within the discipline [1–4]. To engage a diversity of individuals in computing, we need to better understand how to support students in forming positive and meaningful identities with computing. This study centers on a community of African American and Latina teenage girls as they began to form self-conceptions with computing and dance during the MoveLab. The MoveLab was a five-day STEAM (Science, Technology, Engineering, Arts, and Mathematics) workshop where students, dancers, choreographers, computer scientists, and engineers collaborated to create a technology enhanced dance performance. Throughout the workshop, we observed students form and reject self-conceptions [5] around computing. We found that supporting diverse roles of participation, creating a trusting and engaging social environment, having leaders model behaviors, and uncovering preconceived ideas were important characteristics for supporting self-conception formation around computing and dance.

[1] James DiSalvo, B. et al. 2011. African American men constructing computing identity. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), 2967–2970.

[2] Katz, S. et al. 2003. Gender and race in predicting achievement in computer science. *Technology and Society Magazine, IEEE*. 22, 3 (2003), 20–27.

[3] Margolis, J. 2008. *Stuck in the shallow end: Education, race, and computing*. MIT Press.

[4] Margolis, J. and Fisher, A. 2003. *Unlocking the clubhouse: Women in computing*. MIT Press.

[5] Steinberg, L. and Morris, A.S. 2001. Adolescent Development. *Journal of Cognitive Education and Psychology*. 2, 1 (2001), 55–87.

Targeting Intervention Using A Grammar For The Triage of Questions in an On-line Discussion Forum

Nickolas Falkner and Claudia Szabo
The University of Adelaide
nickolas.falkner@adelaide.edu.au

Abstract:
On-line discussion forums are an excellent opportunity for students and staff to discuss issues, ask questions and interact. However, the rapid growth of comments quickly turns an advantage into a burden, when staff have limited time to moderate, influence and respond. How do we identify issues in our student forums in a timely fashion with limited resources? The medical model of *triage* is useful as we want to identify: discussions that are not in need of any help, discussions where we can help, and discussions that are effectively beyond help, for administrative action.

A great deal of work has already been conducted on analyzing forum content and post connections, in terms of semantic content and sentiment analysis, but these methods have varying levels of accuracy and may not give us the ability to take timely action. Graph analysis of discussions based on social network analysis is valuable but we are faced with questions regarding whether student forums are actually social networks and how we can immediately select the most 'needy' forum and thread for our attention. Different social networks have different graph characteristics, hampering analysis and affirming that raw structure is not enough to separate 'active' from 'problematic' from 'burning' [1].

We are developing a framework based on a grammar of student questions, where structural events are classified in conjunction with their graph/time relationships and then mapped to tokens. We aim to reduce the produced token string through grammatical rules to produce terminals that clearly indicate where to rank the thread to achieve the best outcomes for intervention.

[1] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Robin Dunbar. 2013. Dynamics of personal social relationships in online social networks: a study on Twitter. In *Proceedings of the first ACM conference on Online social networks* (COSN '13). ACM, New York, USA, 15-26.

Pedagogical Content Knowledge of Veteran Teachers New to CS

Aleata Hubbard
WestEd
ahubbar@wested.org

Abstract:
In the United States, computer science (CS) high school certification is fraught with inefficiencies that often result in teachers trained in other subjects teaching CS courses. Veteran teachers new to CS are likely to vary in their mastery of the critical types of knowledge needed for effective teaching, or their pedagogical content knowledge (PCK). In this lightning talk we will present our initial efforts to understand the nature of teachers' CS PCK development during their participation in a multi-year, on-the-job training program.

As part of the NSF-funded study *Developing Computer Science Pedagogical Content Knowledge Through On-the-Job Learning*, we are conducting six case studies with instructional teams participating in TEALS (http://tealsk12.org/). TEALS is a non-profit program that recruits and trains tech professionals to work with high school classroom teachers and deliver CS courses. Given the often tacit and multifarious nature of teacher knowledge, we employ multiple data collection procedures to gather evidence of PCK development. Using think-aloud interviews aligned with lesson topics, we are eliciting evidence of teachers' beliefs about teaching and learning, their knowledge of student understanding, and their content knowledge. Lesson observations and questionnaires provide us with information about the effectiveness of various classroom designs and instructor roles employed in this training model. We hope this research will help us identify the critical components of on-the-job training for high school CS teachers and inform the design of future professional development opportunities.

Computer Science is "Hard": Looking at the Gender Gap Between Two Computing Programs

Brittany Ann Kos
ATLAS Institute
University of Colorado Boulder
Brittany.Kos@colorado.edu

Abstract:
We will be comparing two computing programs at the University of Colorado Boulder, a multidisciplinary program in Technology, Arts, and Media (TAM) and a traditional Computer Science program. Both of these programs have large enrollments (800 and 1100 students in 2014-2015 for TAM and CS, respectively), but different demographics. The TAM program has over 60% women enrolled; while CS only enrolls 25% women. In an ongoing study, we are investigating the introductory classes for these programs, in an effort to gain insight into why these they attract and retain such distinct populations.

We present a preliminary analysis that scrutinizes the way material is presented in class. Through ethnographic observation, we have studied classroom environments and discourse, and discovered a large discrepancy in the way students, professors, and teaching assistants refer to the course material.

The CS classes acknowledge the difficulty of the material being covered and have normalized this type of discourse in the classroom. These classes have common phrases, such as "This is difficult" or "This isn't intuitive", which are usually referring to CS in general or the class overall. Alternatively, the TAM instructors almost never indicate how difficult the material might be, but instead focus on how hard the students will have to work to succeed. When students talk about their hardships, they refer to specific lessons or projects they are doing, not computing in general. This is the first phase in a larger study, investigating the effect this discourse may, or may, not have on student enrollment.

Middle School Students Learn Programming Using Wearable Technologies

Dr. Victoria Lentfer
University of Nebraska Omaha
vlentfer@unomaha.edu

Abstract:
The University of Nebraska - Lincoln and the University of Nebraska at Omaha are partnering with the Nebraska Department of Education to develop, evaluate, and deliver the adoption of a project-based curriculum for middle school formal and informal educators in the content areas of computer programming and engineering. This project utilizes wearable technologies as an instructional context by using microprocessors to introduce the basics of engineering and computing using the Arduino platform. Wearable technology is combined with smart textiles and computer circuitry so that the students can immerse themselves in this hands-on project-based curriculum. This project makes a significant contribution to computer science education by establishing the effectiveness of project-based learning in the study of integrating STEM disciplines of computer programming, engineer design, and electric circuitry. The project has the potential for national adoption, for a direct impact on computer science educational practices, reaching traditional underserved populations and for advancing knowledge in the computer science discipline.

[1] Eric Mazur. 2011. The scientific approach to teaching: research as a basis for course design. In *Proceedings of the seventh international workshop on Computing education research (ICER '11)*. ACM, New York, NY, USA, 1-2.

Storyteller: Telling Stories by Reflecting on Code Evolution

Mark Mahoney
Carthage College
mmahoney@carthage.edu

Abstract:
Most significant pieces of software evolve from one small idea into a combination of more ideas, bug fixes, and optimizations. The growth from small to large and simple to complex is a journey. These journeys often have interesting stories that go along with them.

Code comments are usually not an ideal place for telling these stories because some important context may be missing from the final state of the code. It doesn't make sense to document in the comments why an important code refactor was made if there is no evidence left of it in the code itself. These changes may have some valuable lessons in them but there is usually not a good place to write them down for others to learn from.

This lightning talk will describe a tool called Storyteller. Storyteller captures the keystrokes and file operations during multiple development sessions and allows a student to play them back in an animated recreation of the code. The student can then add a narrative to the playback of code over time rather than in the code comments. These narratives, or *stories,* are valuable for students and instructors to explain code evolution over time.

The stories that can be created from the tool can:

- allow a student to reflect on how their code evolved to look for key mistakes and good decisions.
- help a student explain the evolution to a grading instructor. This will help the instructor understand the final product and get a sense of why the code is the way it is.
- help a grading instructor provide feedback by allowing them to create their own stories to highlight the good, the bad, and the ugly.
- help students learn from their peers by watching how others write their code and hearing their stories.

Gender Differences in High School Students' Decisions to Study Computer Science and Related Fields

Jason Ravitz
Google
ravitz@google.com

Abstract:
Computer science is vital to our future -- from the health of nations to an individual's ability to actively engage in the technology that is now embedded in nearly everything we do. Increasing women's participation in computer science (CS) is a critical workforce and equity concern [1]. Researchers, practitioners, and the technology industry have committed to reversing negative trends for women in CS as well as in engineering and information technology "computing" fields [2]. Building on prior research, this study identifies factors that influence young women's decisions to pursue CS-related degrees and the ways in which these factors differ for young men. We conducted a study of 1,739 students across the U.S., surveying high school students and college grads. Using factor analysis to group similar variables into broader factors and logit regression to rate importance of factors, we found that, for high school girls, 95% of the decision to pursue computer science or related fields is comprised of factors that can be influenced: social encouragement, career perception, academic exposure, and self-perception. Encouragement and exposure were particularly huge for girls compared to boys -- with encouragement from parents and other adults significantly more influential. And, for boys, career perception was a larger influencing factor. Self-perception was also important, with no significant difference between both genders. Overall, the most heartening outcome of the study is the limited role that uncontrollable factors play in influencing the pursuit of a CS degree. For high school girls, uncontrollable factors like household income and ethnicity contribute only 4.9% to the explainable factors. The factors most related to female participation in computing fields are actionable. Our presentation will share results of ongoing work to better understand the differences in factors that influence men and women in order to help achieve gender parity.

[1] National Science Foundation. (2012). *Science and Engineering Indicators 2012*. Washington, DC: Author. Retrieved from http://www.nsf.gov/statistics/seind12/c0/c0i.htm

[2] National Center for Women in Information Technology (2015). *Top 10 Ways Families Can Encourage Girls' Interest in Computing*. Boulder, CO: Author. Retrieved from https://www.ncwit.org/resources/top-10-ways-families-can-encourage-girls-interest-computing/top-10-ways-families-can.

How to Train Your Robot:
Computing Introduction for 8-10 Year Olds

Presenter: Chad Williams
Coauthors: Emtethal Alafghani, Antony Daley Jr., Kevin Gregory, and Marianella Rydzewski
Central Connecticut State University
cwilliams@ccsu.edu

Abstract:
Introducing technology to young students has become a critical objective of education programs in today's world. Educators and parents alike are seeking innovative ways to introduce young students to the important skill of computer programming. Within this study, a web application was developed to introduce young students to programming concepts in a fun way through problem solving. By acting as robot trainers, students are introduced to concepts such as: automation, conditions, repetition, and debugging.

The goals of the study were twofold: first, introduce students to how computers work in an enjoyable way and, second examine how they used the introduced programming concepts to solve problems. Two sessions were conducted with eight to ten year olds to evaluate the application. The first session introduced the students to programming concepts in a structured, incremental fashion. The second session challenged them to apply what they learned to create more free form solutions. In addition to surveying participants to understand their knowledge and motivations before and after playing the game, the team also collected analytical data during game play.

This study reports on the trends that were observed in: the programs students developed, debugger use, and variation in the number of attempts and completions between the two sessions. Our results showed the game effectively familiarized eight to ten year old students with both computer language constructs and the essentials of algorithmic thinking, while also increasing the students' own perceived knowledge of computers in a way they enjoyed. Students were quickly able to learn the core programming concepts introduced by the game and apply these concepts to free form solutions. The results demonstrated both males and females enjoyed the game and developed an enthusiastic interest in learning more about computer programming.