SIGCSE 2015 Trip Report - Chris Wilcox

3/10/2015

I just returned from the Special Interest Group on Computer Science Education (SIGCSE), held from 3/4-3/7 in Kansas City. Over 1300 faculty, instructors, and secondary teachers attended the conference. Many attendees are teaching faculty or are doing active research in Computer Science education, or both. The conference continues to be valuable, especially for faculty that teach introductory courses. I start this report with an executive summary, followed by a survey of some of the resources that are now available to CS instructors, and a description of the best paper. Next I talk about possible research and funding opportunities. I conclude with a few of the highlights on the 18 technical papers, 2 birds of a feather, and 2 keynote speeches that I attended. Please send any comments to wilcox@cs.colostate.edu.

EXECUTIVE OVERVIEW

1) The first theme is a growing emphasis on CS education for non-majors. CS departments and other entities such as NSF are saying that CS training is becoming required for a greater number of undergraduates, not just CS majors. As a result, many institutions are ramping up their non-major courses, and a number of papers discussed the specific needs and challenges in teaching CS to this population.
2) The second theme is an emerging recognition of the many online resources becoming available for CS instruction. In this category I include online textbooks, programming tutorials, MOOCs, bulletin boards, assignment repositories, commercial and open source software for automated grading, etc. See the next section for a list and short description of some of the most interesting offerings.
3) The third theme is the continuing emphasis on pedagogic techniques such as the inverted classroom, peer instruction, and pair programming. These are widely believed in the SIGCSE community to increase student engagement, performance, and retention. Many papers were shown that validate the effectiveness of these techniques, including the best paper, which I describe below.
4) The fourth theme is diversity, which continues to be a problem for many departments. One of the most interesting studies I have seen was funded by Google and reported on in the student poster contest. I have attached the white paper from which the poster was produced, in PDF format.

In general it seems that a large number of institutions, including many of the top schools, are increasing their focus on undergraduate teaching. As a result they are doing better at minority and female participation and retention, and in educating non-majors than we are. The participation of our fellow Colorado universities at SIGCSE was impressive, with Colorado School of Mines sending 3 faculty and 4 graduate students, and CU sending at least 3 faculty and several graduate students. I talked with faculty from both institutions: Dirk Grunwald, Judy Stafford, Liz Boese from CU, and Tracy Camp from Mines. We discussed having an annual meeting with faculty from each of these institutions that are interested in undergraduate education, and we are invited to Mines this summer, something I will follow up on.

ONLINE RESOURCES

ONLINE TEXTBOOKS: Zyante (Zybooks) and Pearson (Revel) continue to be leaders in this area, and I attended supporter sessions for both companies. Zybooks is adding automated grading to their online textbooks that will support longer, more complicated assignments. I was able to get a demo from their CTO. The product looked good though obviously preliminary, but they say it will be in Beta in the fall. We may have the opportunity to give them

some feedback to improve their product, if we want to. Revel is a framework from Pearson that is greatly improved from the TuringsCraft software that we previously evaluated. The Revel course for introductory CS is based on the Daniel Liang Java textbook. I looked at the textbook and slides and found them to be very good, even without the online component. The textbook appears to cover all of the material we are considering for our CS1 and CS2 courses.

AUTOMATED GRADING: Web-CAT remains the leader in this area. I talked with Stephen Edwards about signing up to evaluate the software on their server, instead of installing it here. Web-CAT is open source and freely available. As mentioned, Zyante is also becoming a player in this arena. Revel does not appear to support real program grading, just scoring of snippets. In addition there is a new player called Vocareum, which offers a pure automated assessment platform (including program grading). I was unsure that their business model would hold up since they are similar in price to Zybooks and Revel, without any content. Dirk Grunwald from CU also introduced me to several open source packages that he is looking at for program grading. As a result of looking at the competitive landscape for automated grading software, I have decided not to pursue doing a startup in this area.

BULLETIN BOARDS: Piazza really dominates this category, there was lots of talk about how much this (free) product enhances communications in classes. Piazza can be integrated into any website that is based on PHP, I talked with one of their developers about this.

ASSIGNMENT REPOSITORIES: Repositories offer (free) pedagogic materials and assignments for teaching CS courses. This category is really growing, the more interesting repositories include CS teaching tips from Harvey Mudd (http://www.csteachingtips.org), problets from Amruth Kumar sponsored by NSF (http://www.problets.org), and Nifty Assignments from Stanford (nifty.stanford.edu).

BEST PAPER

The best paper was called *Structuring Flipped Classes with Lightweight Teams and Gamification*, from the University of North Carolina in Charlotte. This paper was definitely deserving. The premise of the authors is that 1) the first computer science class is very important in determining the attitude and success of undergraduates, and 2) most freshman are scared and isolated, which impacts their academic performance, and 3) that building community in the first class will greatly improve student engagement and therefore academic performance and retention, and 4) that this is especially important for female and other under-represented students. They therefore made a radical modification of their CS1 course, as follows:

First, they inverted the classroom and required students to complete the reading, associated exercises and videos, and online quiz to make sure the outside work was being done. Second, they created a 3 hour programming lab where students were divided into teams of 5. They spent the entire semester with these teams. The presenter said that they view this as “handing each freshman four new friends on a platter”. Programming exercises in this were low stakes, thus having a team that was not as strong did not inordinately penalize a student. Finally they have a weekly 1.5 hour help session with the instructor in which a number of activities (other than lecturing) were done. In addition they had an elaborate system of rewards and penalties based on 3D printed badges in which students good earn 1% grade increments by participation, asking good questions, and extra effort. Interactions were encouraged both within and between teams. They tracked the performance of each group and published a leader board. In addition they used a variety of relevant and compelling assignments, including game programming.

The results were amazing in terms of student satisfaction and performance. Students reported a much higher sense of belonging, and they performed better in a variety of areas. The most convincing proof of the effectiveness of the approach was their comparison of academic performance between the control and experimental group on

the successive CS2 course. The students in the experimental group had a huge increase in grades versus the control group, and (in absolute percentages) 32% compared to 18% A's, 41% compared to 38% B's, 16% compared to 26% C's, and fewer D's and no F's! They are currently measuring retention and some other factors, while continuing the new organization of CS1 and extending it to the CS2 course.

RESEARCH OPPORTUNITIES

My SIGCSE paper this year was called "The Role of Automation in Undergraduate Computer Science". It had a good reception, and with the help of Noah John I was able to perform a survey of automated grading duing the session. Many of you saw my practice talk, so I will not repeat any of what was presented. Next year I plan on submitting to SIGCSE a paper called "The Science of Automated Grading for Student Programs". The main topic of this paper will be a discussion of the difference between regression testing, as used in industry, and program grading. I plan on enumerating some of the innovative strategies we are currently using, as well as discussing other issues such as how to detect the state of student programs that are unresponsive or in an infinite loop.

I am considering applying for funding my research from NSF so I talked with Paul Tymann (ptymann@nsf.org) who along with Mike Ehrlinger are the program directors that attended SIGCSE. There are two divisions of NSF that primarily sponsor CSE research. One is the Division of Undergraduate Education (DUE) in the EHR directorate. They currently have an initiative on Improving Undergraduate Science Education (IUSE), which includes all STEM areas such as Computer Science. Grants can be exploratory, i.e. proof of concept or design and implementation. There's about to be a new solicitation, but he said it will be similar to the previous one. The other initiative that might be applicable is CyberLearning in the CISE directorate.

CONFERENCE HIGHLIGHTS

TECHNICAL SESSIONS

I attended six technical sessions, each with three papers. These included *Education Research, Automated Assessment*, *CS1/CS2, Student Engagment/Active Learning, Non-Majors/Interdisciplinary,* and *Teaching Practices*. Some of the major highlights were as follows:

- Noah John did a very creditable job on his and Jaime Ruiz's talk on *Student Response to Teaching of Memory Cues and Resumption Strategies in Computer Science Classes*.
- *Closing the Cyberlearning Loop* from the University of Colorado was an interesting presentation on the real-time detection of students that are in trouble to allow more timely intervention.
- *Reconsidering the Impacts of CS1 on Novice Attitudes* from DePaul Univ. discussed the benefits of offering separate courses for those with and without programming experience.
- *Drop, Fail, Pass, Continue: Persistence in CS1 and Beyond in Traditional and Inverted Delivery*, from the Univ. of Toronto talked about improving the performance of weaker student by inverting the classrom.
- *Reconsidering Automated Feedback: A Test-Driven Approach* was a Web-CAT paper on a complex reward system for teaching students to use test-driven development.
- *Improving Non-CS Major Performance in CS1* from Calvin College claimed that moving their non-najor course from C++ to Python slightly improved student performance, but not as much as expected.
- *Evaluating Pair Programming for Non-Computer Science Major* Students claimed that pair programming significantly improves engagement, attendance, and academic performance, even for non-majors.
- *What Influences CS Faculty to Adopt Teaching Practices* was a fascinating exploration of the factors that encourage or discourage the adoption of new pedagogic practices.

- *Personalized Attention @ Scale* was the paper from the Univ. of Colorado in which Liz Boese presented their novel approach to "evaluation grading" as previously reported in my report on our CU visit.

BIRDS OF A FEATHER

The first BOF session I attended was called *Automatically Generated Feedback for CS Student Work: Best Practices* chaired by faculty from Drexel University. This is more or less the same as previous BOF groups on automated assessment. There was usual admission that most faculty prefer to roll their own automated grading, and this was bemoaned. There was the usual desire for a central repository for assignments and tests, which will probably never happen. Besides that there was lots of enlightening conversation of the advantages and drawbacks of automated assessment.

The second BOF session I attended was called *Working with Undergraduate Teaching Assistants: Best Practices and Lessons Learned*, chaired by Chris Gregg (Tufts University) and Colleen Lewis (Harvey Mudd College). This was basically a session in which everyone celebrated the benefits of using undergraduate teaching assistants, and how to use them most effectively. I think we're well-positioned on this issue, as most institutions are doing something similar to what we do, except in some cases they don't even pay their undergraduates!

KEYNOTE SPEECHES

Educating for Both Art and Technology was presented by Jessica Hodgins from Carnegie Mellon, who is also a vice-president for Disney Research. I don't have much to say about this, other than you should see the amazing quality of animations that can be produced by a combined class of art and computer science students from Carnegie Mellon. The point of her speech was that in the future cross-disciplinary instruction will be increasingly important.

Data Structures Courses: Past, Present, and Future was presented by Mark Weiss, from Florida International University, who received this year's SIGCSE award for Outstanding Contributions to Computer Science Education. This talk was entertaining but I didn't bring much away other than that data structures can be taught in any language. The presenter proposed teaching algorithms and computing using Excel spreadsheets!