SIGCSE 2016 Trip Report - Chris Wilcox

3/8/2016

I just returned from the Special Interest Group on Computer Science Education (SIGCSE), held from 3/2-3/5 in Memphis. Around 1300 faculty, instructors, and secondary teachers attended the conference. Most attendees are teaching faculty or are doing active research in Computer Science education, or both. I have condensed some of the more important points into an executive summary, followed by a more detailed report of the presentations, papers, workshops, and highlights from the exposition. Please send any comments to wilcox@cs.colostate.edu.

EXECUTIVE OVERVIEW

1) Diversity continues to be a major focus of the conference. Many institutions have significantly improved female participation, but minority representation remains a problem. The success factors for addressing underrepresentation are now well known: a sustained effort with administration and faculty support and accompanying resources for outreach, mentoring, and curriculum changes are needed to be successful.

2) Online materials for computer science education continue to grow, including extensive online curricula, learning management systems, and automated grading tools. Many institutions are using online textbooks to replace traditional textbooks, and a variety of resources for K-12 instruction (or outreach) are available. See the section on online resources below for more details.

3) Increasing enrollments are problematic at many institutions, and adjunct faculty are in short supply. I attended a BOF session for adjunct faculty where over 30 attendees said they had (often more than one) open positions. It was claimed that 9 openings exist for every qualified candidate. Factor in salary, prestige and workload concerns, and the result is a shortage that may impact our hiring.

4) Emphasis continues on pedagogic techniques such as the inverted classroom, peer instruction, and pair programming, which are now widely accepted and well researched. Many departments are using the enrollment boom to increase staffing and resources to address diversity issues and to improve and grow service courses such as introductory programming or data science courses for non-majors.

ONLINE RESOURCES

ONLINE TEXTBOOKS: Zybooks (Zyante, www.zybooks.com), Revel (Pearson, www.pearsonhighered.com/revel), and MyProgrammingLab (TuringsCraft) continue to be leaders in the area of online content, but there are many others. I’m still comfortable with the replacement of our Savitch textbook and Zybooks tutorial with the online Liang textbook delivered online via Revel, but we should carefully evaluate the effectiveness next semester. I had the opportunity to go out to dinner with Dr. Liang, and I was very impressed.

AUTOMATED GRADING: Web-CAT remains a popular solution, but a number of open source and commercial options are emerging including GradeScope (www.gradescope.com), Vocareum (www.vocareum.com), and CODIO (www.codio.com). The GradeScope product is very similar in scope and philosophy to our automated grading and looks like the best option if we decide to use commercial software. The product is based on a Linux container call a Docker (www.docker.com) that essentially lets you build a Linux image for a lightweight virtual machine with associated tools and compilers. Building automated graders is very simple with the product, you just create an archive file with test scripts, programs, and data.
Vocareum remains an enigma, they have very little content, yet their product is expensive and their automated grading seems primitive to me. CODIO combines (somewhat limited) content with a learning management system and an intermediate level of automated grading, their goal is to provide a learning management system for computer science. I have a brochure from CODIO if anyone is interested.

K-12 EDUCATION

I investigated what instructors are using for outreach activities, primarily for middle school and high school students, and there are many different solutions including Scratch (scratch.mit.edu), which is still seen as the best solution for elementary school and middle school students without previous programming experience, Khan Academy (www.khanacademy.org/computing/computer-science), which partnered with a couple of Dartmouth professors to create introductory computer science materials, Alice (www.alice.org), a 3D programming environment for novices, CodeHS (codehs.com), which has web-based curriculum specifically targeted at K-12 students, and code.org (code.org), which sponsors the Hour of Code and other K-12 curriculum.

OTHER TOOLS: Piazza (www.piazza.com) continues to dominates the bulletin board category, as GitHub rules the source control segment, at least for academic projects and teaching. Assignment repositories continue to proliferate, including offerings from Harvey Mudd (www.csteachingtips.org), problets from Amruth Kumar sponsored by NSF (www.problets.org), and Nifty Assignments from Stanford (nifty.stanford.edu).

WORKSHOPS

The first workshop I attended was “High Yield in the Short Term: Planning Strategically to Get Women into your Major” by two women that are very involved in NCWIT, Lecia Barker and Leisa Thompson. Much of what was discussed is similar to the plans outlined in our diversity slides, including K-12 outreach, changes to make curriculum more relevant for female students, role models, atmosphere, mentoring, etc., but one thing that we haven’t thought about is the idea of advertising the major on-campus. With increasing enrollments I’m not sure we want to do this, but they found that such advertising (leaflets, department tours, information sessions) was effective at broadening participation, particularly if you can find venues where female STEM students are found, for example biology, engineering, and math. I received a vast amount of information on recruiting and retaining female students from NCWIT, come by if you’re interested. An another technical session I saw a presentation which documented the significant involvement of females during the early history of computing. The speaker claimed that the view of computing as male-dominated appeared during the 1970’s, and is driven more by media representation than reality. Lecia told me that the downturn in female participation since the 1980’s is widely believed to have been caused by the introduction of the personal computer!

The second workshop I attended was “How to Plan and Run Summer Computing Camps – Logistics” which was presented by faculty from several institutions that have been running such camps for several years. The material included recruiting and advertising, curriculum choices, staffing, and a number of pragmatic suggestions for how to make programming camps successful. This was very timely since we’re in the process of planning summer camps.

DEMO SESSIONS

I attended “Creating and Grading IPython/Jupyter Notebook Assignments with NbGrader”, and was very impressed with the approach and implementation. The framework is targeted at Python courses, for which it provides a combination of Wiki and interactive programming tutorial. I attended this demo because Chuck Anderson has been using it and I was intrigued by the positive feedback from a couple of his students.
I attended the following panels:

“Rediscovering the Passion, Beauty, Joy, and Awe: Making Computing Fun Again” – This panel thinks we should focus on problem solving instead of syntax, and they reviewed the many online curriculum options as described in Online Resources above.

“Why Don’t Some CS0 Students Succeed? How Important are Background, Experience, Culture, Aptitude, Habits, and Attitude.” This panel discussed whether a ‘geek gene’ is required to be good at CS, as many faculty (including Donald Knuth) believe, or whether it’s just hard work and persistence, as the panel members believe. They mentioned the segregation of students with and without programming experience as a success factor in broadening diversity, and this is from Daniel Garcia at Berkeley who teaches an introductory course which is 50% female. They stated the common reasons for failure - lack of time, minimal contact with instructors and staff, and the common attributes of students that succeed - logical thinking, problem solving, persistence, love of tinkering, abstract thinking, lack of fear, sitting in front, etc. I don’t think anyone would argue with either list.

“Booming Enrollments – Survey Data” was dominated by Tracy Camp at the Colorado School of Mines. She put some numbers on the increases we are all seeing. One reason for higher enrollments may be that more K-12 students are getting exposure to the field, because of the efforts of code.org and other organizations. She also talked about the new CS principles course for K-12. The panel believes that this boom will last longer than the previous enrollment spikes in the mid-1980’s and early 2000’s.

“igniteCS: Addressing Undergraduate CS Retention” was moderated by a Google employee. Several schools that received grants presented their experiences, which mainly involved outreach programs such as summer camps, and community building revolving around ACM-W chapters. An interesting statistic presented is that 80-90% of participants in Girls Who Code clubs end up majoring in CS. One institution mentioned that there was some backlash from the ACM club when they founded the ACM-W chapter, but they were able to resolve this through collaboration. One of the summer camps run under igniteCS was project-based, with 3D printing, virtual reality, and Scratch programming. Another started with Scratch, then moved to Arduino devices, and finished with Python. One presenter mentioned that some female students feel “patronized by block languages”, which is interesting because I have encountered the same attitude in my Girls Who Code course. Overall the institutions that received grants clearly did an enormous amount of work, which required lots or organization, so I don’t think you want to enter into igniteCS lightly.

I attended a number of technical papers, here are a couple of highlights. Liz Boeze presented “Just-In-Time Learning for the Just Google It Era”. She maintains that since students have access to Google and StackOverflow, we need to focus more on problem solving, and less on syntax and knowledge that is widely available. More specifically, she gives open-book, Internet-enabled exams. She also has incorporated assignments from biology and other fields to make the material more relevant to female students. Beth Simon had the best paper entitled “A Multi-Institutional Study of Peer Instruction in Introductory Computing” which continues to proselytize peer instruction. My paper “Testing Strategies for the Automated Grading of Student Programs” seemed to be received well.
KEYNOTE SPEECHES

Cognitive Load Theory and Computer Science Education was presented by John Sweller from the Univ. of New South Wales. This talk was esoteric, but had some interesting ideas, including the fact that our short-term focus is very transient, and that our goal must therefore be to facilitate moving knowledge into long-term memory where it can be applied again and again. The speaker said that students experience lectures very differently than the instructor does, because the material is new to them. Instead of expecting students to make the leap to new concepts, he suggests that showing working examples first and having them repeat them is more effective.

CS Education: Catching the Wave was presented by Jan Cuny, the Program Director for Broadening Participation in Education in Computing for the NSF. This talk was fascinating and very informative. The goal of having 10K teachers of CS in high school by 2016 was obviously missed, the actual number is more like 2K, but they hope to double that in the next year or two. She also discussed the challenges of increasing diversity while trying to cope with enrollment increases. One amazing statistic was the number of high school students that various AP exams. All of the AP fields have grown enormously, except Computer Science, which lags far behind.