Future Directions in Computer Science Education Research

Steve Cooper
Mark Guzdial
Beth Simon

ACM Ed Council
November 3, 2013
What I’m seeking your help to do

• Help us to identify the topics we should be discussing at the summit in January
  – You’ll be breaking up into groups to discuss general topic areas, and to make recommendations
  – You can also make suggestions about specific expertise we should have at the summit
Background

• Older history of CS education
  – SIGCSE (1970)
  – Papert, Soloway, etc. 1980’s

• More recently
  – Bootstrappers, Scaffolders (2001 – 2004), and Bracelets (2005)
  – ICER
The status quo (in the US)

• Faculty doing CS ed research
  – At doctoral and non-doctoral schools
  – In schools of Education

• Students doing PhD theses in CS ed
  – Limited: Education students working with Roy Pea/Brigid Barron (at Stanford), or those working with Aman Yadav (Purdue)
  – Limited: CS students working with Mark Guzdial (Ga. Tech), and other “one-off” students
  – Limited job opportunities
The purpose of a CS education research directions workshop

• Identifying the major research areas our community should be investigating over the next 5+ years
  – Helping inform NSF (for future solicitations/funding decisions)
  – Similarity to what the DB community does

• Strategizing to get CS Ed research accepted as a valid sub-area of CS
  – Hiring CS Ed researchers at doctoral and non-doctoral institutions, and supporting them
  – We’ll deal with this issue in a later workshop (interested in being involved?)

Red = to be addressed in a future workshop
Process

• Requested 3-5 page white papers suggesting future research directions for CS ed
  – Received > 100 white papers with ~ 300 authors for 30 places at the workshop
    • From CS ed researchers, learning scientists, education researchers (e.g. TERC), and Engineering education researchers
    • CS Ed Research does not mean (to us) "Research into how to make our current courses more effective."
      – Real education research doesn't presume that our existing curriculum and existing classes are the right way of carving it up.
Anticipated major themes

- Broadening participation
- K-12 issues
- Tools
- Students/learning
- STEM education
- Structure/Policy

Note:
Red = I expected to see this topic, but did not
Blue = I saw this topic, but didn’t expect to see it
Broadening participation

• Women
• Under-represented minorities
• Accessibility
K-12 issues

• Teachers and professional development
• Curriculum
• Outreach
• Informal education
Tools

- Algorithm visualization
- Robots
- MOOCs
- Introductory CS tools
Students/learning

• Assessment
• Pedagogy (POGIL, PLTL, SI, PI, etc.)
• Community college
• Topics
  – Parallelism
  – Security
  – Software engineering (including producing good Ses
  – Interdisciplinary programs
  – Data science
• MOOCs
• Measurement of learning/analytics
STEM Education

• Intersections between CS education and X education ($X \in \text{SEM}$)
• Lessons to be learned from other X education
• Incorporation of CS content/instruction into other classes (especially in K-12)
Structure/Policy

• Developing Research-Informed Instructional Resources to Support Broad Scale Access to High Quality Computer Science Education
  – A meta-analysis of the needs of K-12 CS ed
• Policies to help build institutional capacity
• Evaluating the current state of knowledge within the field of computing education research
Other themes?

• What are we missing?
Task

• Choose a group (students/learning might need to be 2 groups)
  – Identify which sub-topic areas seem to be most important
  – Fine to identify specific research questions/agendas

• Slides available at:
Reports from the groups

• Broadening participation
• K-12 issues
• Tools
• Students/learning
Broadening participation

*Barbara Owens, Jane Prey, Alison Clear, and Lisa Kaczmarczyk*

- What are ways to frame questions to get academics to value CE research?

- Questions must address the stakeholders such as NCWIT, WIC, as they interact with academics.

- What are the most compelling questions that will bring together underrepresented groups, women, accessibility community, policy makers and the above groups so that academics value that research?

- The question of gender, accessibility and underrepresented groups might be part of every question that the other groups raised.
K-12 issues

Dan Garcia, Deborah Seehorn, Chris Stephenson, Steve Cooper

• What are the indicators of incoming student success in introductory level computer science in colleges and universities?

Does computer science learning in high schools contribute to success/improvement in other disciplines, especially mathematics and science?

What is the link between age/educational development and the potential to learn and master computer science concepts?

Are there issues of ergonomics in the introduction of computing devices with young children?

Is there a link between previous math learning and success in computer science at the high school level?

What are the major factors that determine students making early choices not to pursue computer science?
• What is the role on informal education programs in scaffolding learning in computer science, especially in communities where access to computer science learning in school is limited?

What are the potential benefits and drawbacks of MOOCs in middle school and high school student learning?

What are the potential benefits and drawbacks of MOOCs for the professional development of computer science teachers?

What models professional development are most effective for improving teacher mastery of computer science concepts and pedagogy?

What are the impacts of current efforts to market computer science to students?

To what extent does poverty/ lack of home access impact computer science performance and or interest in school?
K-12 issues - 3

• Do one-to-one devices per child programs have any impact on computer science interest or performance?

What are the major factors in computer science teacher retention?

What is required to increase the availability of teacher preparation programs for computer science teachers?

What is the impact of transitioning the the content of teacher preparation courses in "educational technology/AV"to a computational thinking in STEM course?

What is the ideal balance between content knowledge learning and pedagogical learning in computer science teacher preparation and alternative certifications?

What are the effectiveness of hybrid programs (educators and volunteer partnerships) in improving student access to rigorous computer science courses and the increasing the pool of well-prepared computer science teachers?
Tools

Dan Grossman, Peter Norvig, Mehran Sahami

• If logging everything, should we analyze data across students on a problem or across time for a student?
  – Should we focus attention where a problem easier to do research but less illuminating?
  – There really is a continuum between these extremes
  – Analogy with Google learning from web search

  – And another way is controlled lab setting (eye-tracking, etc.)

  – What is the general value of A/B testing?
Tools - 2

• A separate theme is /creating/ better tools (distinct from research on evaluating an exiting tool)
  – auto-grading of all forms (style, hints, tutoring, ...) (emphasis is not necessarily on the /grading/)
  – more general automated assessment and feedback

• Discussion forum as a tool
Tools - 3

• Measuring/promoting/researching tool adoption
• Student-evaluation mechanisms correlate with what?
• The relative value of fast feedback vs. good feedback
• Compilers
  – compiler messages for beginners based on prior corpus
  – bilingual compilers (messages explaining errors in terms of other languages students know)
Tools - 4

• Teaching programming in a [stackoverflow.com](http://stackoverflow.com) world? Has CS Ed been too slow to react to 'social networks' about programming?

• Effectiveness of education tools like LMS, is that even CS Ed research specific?

• MOOC tools (tools for scale, tools for widespread large collaborations, ...)

• Tools to improve, utilize, measure-the-effectiveness of peer assessment
Goals and Motivations:
Where do the students want to go, i.e. what are their goals and motivations for learning within our disciplines? Presupposes that students are best served by being moved along the trajectories that they care about.

More generally, learning goals emerge from many different places. So it is important to be explicit about where goals emerge from, and be sensitive to these different audiences. In other words, research questions might come from a variety of different places.
- students
- teachers
- institutions
- parents
- industry
- ...

Depending on the audience and goal, the research strategy should adapt (e.g. action research for teacher questions, or sociological approaches for policy-related questions).

What are student beliefs about the computing disciplines, and how does this affect their choices about further pursuing one or more of these disciplines?
- How do student conceptions of the discipline change as students engage in the “materials of the discipline”, i.e. as they learn to program, learn to abstract, etc.
Students/learning - 2

• **Linking to everyday or other forms of disciplinary thinking**
  
  – In what ways can pedagogies for programming bridge to everyday ways of “doing computation”, such as following recipes, procedures, and similar. And how might this reflect back on our question about innate capability or not?
    
    • Are there “paradigms of thinking” that students can learn that students can transfer from their other areas of life to programming and computational thinking?
  
  – If we ground students in concrete and contextualized ways of learning, how do they learn more abstract ways of thinking and doing? How do they lift above context and concreteness?
• **Linking to learning and developmental theory**
  – What is informing computing education research in terms of fundamental learning and education theory? Can these be made more explicit? If Computing Education Research (CER) is to advance, it needs to be built from general theory so that it is scientific and can be built upon.
    • This is particularly challenging for teaching practitioners, who often don’t know much about this (e.g. it isn’t part of the education for a PhD student in CS).
  – Are there developmental constraints for learning different kinds of computational thinking at different times in human ontogeny, as has been hypothesized for Math and formal kinds of reasoning, such as what Piaget has asserted.
• **CE-specific questions**
  
  – Do visual programming languages (e.g. Alice, Scratch, Blockly) provide a “natural” or “first” step from which students can learn computational reasoning. Research reaching back 20+ years (Schneiderman, T.R. Green, Steve Payne) indicates “yes” and can be built on for further development. The Psychology of Programming Interest Group (PPIG) in Europe also regularly looks at questions like this.
    • Similarly, there is already research on “what kinds of languages are easier to learn”.

  – What about research in the value of pair programming?
    • Similarly, research on joint or sequential production of programs, e.g. where one student alters the code of another.
• Misc
  – Can anyone learn to program? Is this an innate bar that some cannot clear? And if so, how does this affect pedagogy?
    • this is similar to questions from Mathematics and Science education
  – What are the relationships between “preferences for learning” along the concreteness to abstract ways of thinking and student preferences within the computing disciplines.
  – Will “pedagogical patterns” be helpful as ways to generate instructional designs for CE? Do we have discipline-specific patterns, what Lee Shulman calls signature pedagogies?
  – Will the further penetration of computing in K-12 provide legitimization and wider practice of CER, in the way that it has done for Math and Science Education Research?