Education & outreach

- 2013 summer school
- high school students
- intelligent tutoring software

summer school
- decide location format, date, duration, target audience.
- should have people beyond us
- size? how many participants?
- dedicated local organizers, volunteers

Prior experience?
- do we want to host our own or join existing summer school?
  - dagstuhl?
    - we want to host summer school in US, but dagstuhl format is good
  - size: lots of students. ~100 students (phds)?
    - given the size, it might better to join an existing summer school to better handle logistics. see dagstuhl as an example.
- summer school will evolve too
- we need to make some decision soon.

outreach to high school students
- several programs like cornell upenn michigan.
- “programming is no coding”

Hadas Kress-Grazit
CURIE academy - week long summer camp for high school girls
- program robots using matlab?
- 80% never programmed at all
- many are not physical science side, they are from biology, chemical...

EYH - day of workshops for middle school girls
- using the iRobot Create - program using the bump sensor
- programming is not coding, we might be able to use some of the ideas here.

Upenn outreach program
- summer academy in Applied Science & Tech
  - strong connections to penn CIS
- women in computer science
  - high school day for girls
- dining philosophers/penn apps
  - hackathons
  - bring in experts to teach students some high tech
  - spend an afternoon to develop tools, etc.

Michigan
- spring courses with detroit area pre-college engineering program
• summer engineering academy (middle & high school)
• Q: if we can have the students for the whole summer, we can have participants for user study, tool evaluation, etc.
• Q: illinois also has women in computer science program. they might be able to participate the outreach program.

Intelligent tutoring software
• a component of our knowledge transfer
• target audience: high school and undergrad students

Armando Solar-Lezama: smart graders for programming assignments using synthesis technology

• idea: apply the synthesis tech we have to problems of automated grading.
• like automated bug finding, but very different because you already have the right answer.
• the software quality problem is a symptom of programming needs exceeding the supplies of eligible programmers.
• apply the lessons learned to
  o make programmers more productive
  o make programming more accesible
  o reduce the cost of training the next generation
• test-cases based grading is not very good
  o no precise correctness correlation
  o no student tailored feedback, can’t tell the student how far they are
• manual grading by TAs
  o error-prone, time consuming, not scalable.
• PexForFun website
  o people submit programs to complete some tasks
  o the website give feedback whether your program is correct or not
  o Example: a student keep submitting an almost correct solution, but frustrated by the feedback “your program failed this input”, and cannot find the answer
    ■ they are novice users learning programming
• If the system can provide advice how far you are from the correct answer, then the students can learn more.
• approach:
  o use data of previous student solutions
  o correction rules based on corrections
    ■ a<b  -> a {<= | > | >= | == | !=} b
    ■ a[i] -> a[ { i+1 | i-1 | i-?i} ]
  o Q: can you just measure the distance of student’s program and the correct answer based on syntax info? why use synthesis?
    A: student’s submissions has many syntactical variations, if you want to measure distance, you have to consider semantic meanings.
    and the correction model can be different, the grader can provide a customized set pf correction rules
• transformative for students in under-funded schools
  o reduce the resources required to support quality instructions
• same tech could be used for automatic tutoring
  o identify errors stemming from deep misconceptions
- synthesize small examples that make misconceptions explicit
  - Q: what’s this different compared to program visualization technology?
    A: if you want to run the program visualization backward? the student
    want to ask “if I want to do this, what code changes should I make”? then
    synthesis can help.

Automatically generating problems for online courses in embedded systems
Dorsa Sadigh, uc berkeley
  - recent trend towards massive open online courses
  - generate problems
  - given existing problems, we want to generate similar problems
    - intro to embedded systems
    - problems in the text book
    - example: real-time scheduling
      - studies fixed-priority
      - some of the the problem text can be changed, we can use a
        template text
      - we can formulate solutions
      - generate random numbers to replace holes in template text,
        then use SMT solver to solve the solutions, and we get new
        problems.
      - Example: we have model, property and traces, related one another
        in the 1st problem, you are given <phi> , find <M>; 2nd problem, given
        <M> find <psi>.

implement the ideas for model based problems
bring creativity in automatic problem generation
getting more data from the students this fall.

Q: how to calibrate the difficulty of the problem?
A: if you use the same template, usually the difficulty levels of the new
problems are similar.

Course modules
  - target audience
  - covering the breadth of ExCAPE
    - collaborate between PIs?
  - how things fit in the summer school?
  - master plan?

look forward to involvement of all PIs

Q: combine the summer school and outreach. generate lecture notes in
summer school to use in outreach programs.