K-12 Computer Science: Aspirations, Realities, Challenges and Opportunities (a US perspective)

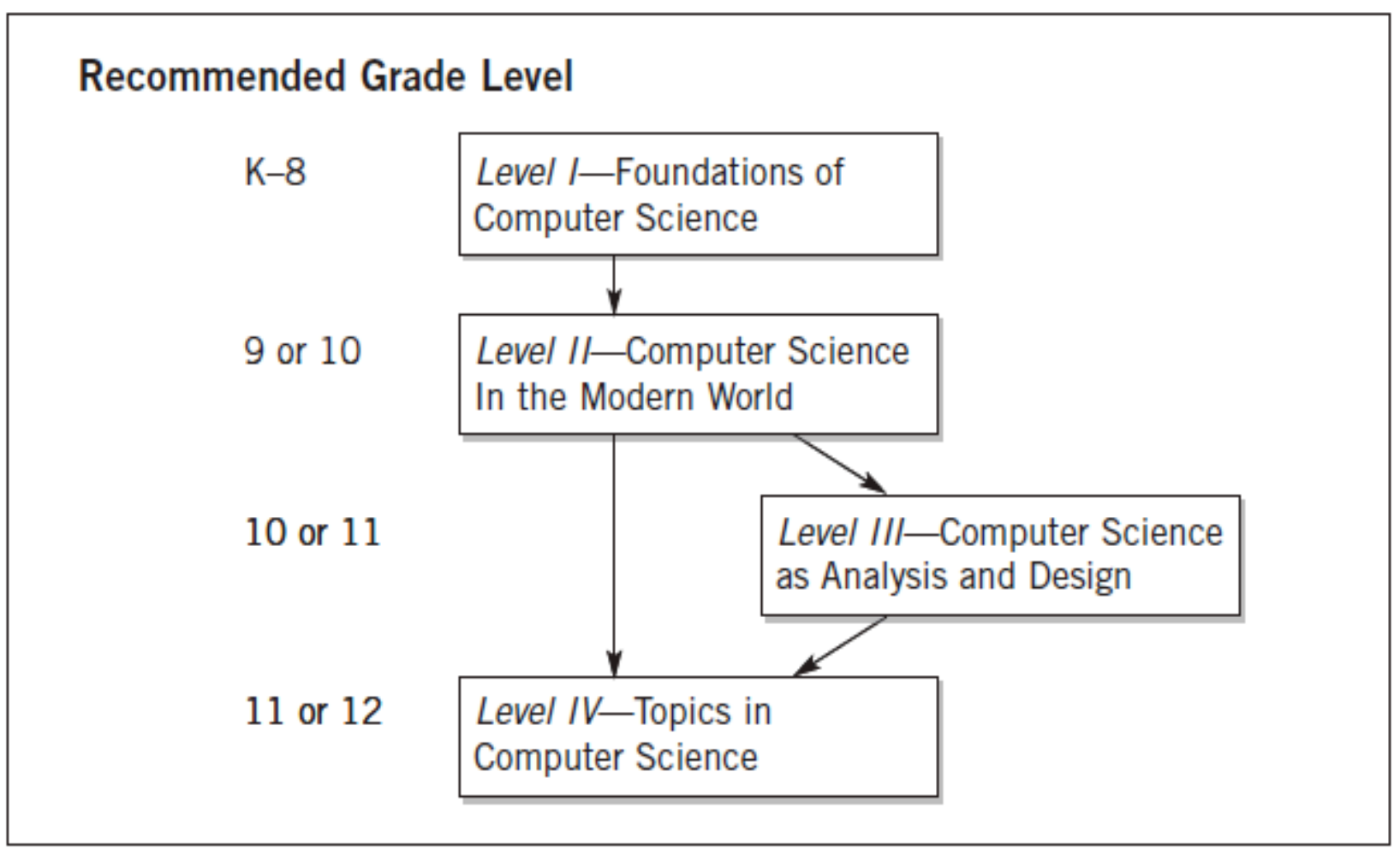
Allen B. Tucker
Bowdoin College
January 2010

allen@bowdoin.edu
http://bowdoin.edu/~allen

The Problem in 2003

- Computer science (CS) was not well-understood by the general public in the US,
- CS concepts and skills were not viewed as particularly important,
- The only coherent K-12 CS activity was the AP curriculum,
- In place of CS, only Information Technology (IT) was offered as a skill to support traditional academic subjects, mainly science.

One Response in 2003: ACM Developed a K-12 Computer Science Model Curriculum (1)



Level I: Grades K-8

Foundations of Computer Science

1. Integrates basic skills in technology with simple ideas about algorithmic thinking.
2. Provides hands-on activities help ensure that students meet these goals.
3. For examples, see http://csunplugged.org

Level II: grade 9 or 10

Computer Science in the Modern World

1. Accessible for *all* students, whether they are college-bound or workplace bound.
2. Includes fundamental concepts of computers (hardware, software, operating systems, etc.), networks, and algorithmic problem-solving.
3. Introduces students to computing careers and ethical issues.

Level III: grade 10 or 11

Computer Science as Analysis and Design

1. A one-year *elective* course
2. Should earn a *math or science credit*
3. Emphasizes the *scientific and engineering aspects* of computer science, focusing on:

    (a) mathematical principles,

    (b) algorithmic problem-solving and programming,

    (c) hardware, networks, and social impact.

Note: this level can include the CS AP course.

Level IV: Grade 11 or 12

Topics in Computer Science

1. Either a *projects-based course* or a course leading to *industry certification*.
2. The Level II course is a prerequisite.
3. Example projects: Multimedia, Graphics, Web Site Development, Animation, Networking, Simulation and Modeling.
4. Example industry certifications: Certified Internet Webmaster (CIW), A+ Certified Technician, and i-Net+.

Aspirations of the K-12 Model Curriculum

- Individual states would develop *academic standards for CS*,
- School districts would begin teaching *CS principles to a large number* of students,
- *New Teaching materials* would be developed to support these new courses, and
- Schools of education and in-service programs would begin *preparing teachers* to offer these courses.

A Second Response in 2003: ACM formed the Computer Science Teachers Association (CSTA)

To promote the development of K-12 CS by:

- Helping to build a community of CS educators

  (e.g., supporting professional development),
- Advocating for the new *K-12 Model Curriculum*

  (and refining each of the four levels),
- Supporting projects that communicate the excitement of CS to the general public,
- Conducting research on CS education, and
- Recommending policies to improve the status of CS in the high school curriculum.

The CSTA Developed Rapidly

By 2009, the CSTA had grown to over 7,000 members.

(high school and middle school teachers, college and university faculty, and industry representatives)

- CSTA established a place for *K-12 teachers at the annual SIGCSE Symposium*.
- CSTA supports an *on-line repository* for K-12 CS teaching materials.
- CSTA supports a *peer mentoring system* to encourage collaboration among CS teachers.
- CSTA spearheads *many other initiatives* to help improve K-12 CS education in the US.

What's Happened since 2003?

Computer Science is the *fastest-growing professional sector* for the decade 2006-2016 (2).

Yet, the percentage of college-bound students interested in majoring in CS *has dropped* from 4.5% 2003 to 2.9% in 2008.

- States have been *slow to develop academic standards* for computer science.
- *A few* school districts have implemented *new courses* that teach CS principles for a wide range of students.
- A *modest amount of new teaching materials* has been developed to support these new courses.
- *Schools of education and in-service programs have not trained significant numbers* of K-12 CS teachers.

Realities: Slow Growth

Results of a 2009 CSTA survey of 14,000 US school districts (5). Only 1153 (8%) districts responded:

- 65% reported offering one or more introductory (pre-AP) computer science courses.
- 44% said the course was required for all students.
- the *content* of what is called "computer science" is typically *information technology skill-building*:
  - ➔ IT skills support the traditional disciplines.
  - ➔ IT gains a business or "tech" credit (not a math or science credit).

Realities: State Responses

- Most individual state academic standards identify computer science as IT.
- IT typically shows up under the heading "Science and Technology."
- State standards differ about the place of computer science in the K-12 curriculum.
- State standards do not acknowledge CS as a core mathematical or scientific discipline.

Example State Standards

Virginia Computer Technology Standards of Learning for Public Schools (June 22, 2005): "The Standards identify and define the progressive development of *essential knowledge and skills necessary for students to access, evaluate, use and create information using technology*."

Pennsylvania Academic Standards for Science and Technology, Pennsylvania Department of Education (January 5, 2002): focuses *exclusively on the use of technology in science education and nowhere else*.

Example State Standards

New Jersey (2009): "In grades 9-12, students *demonstrate advanced computer operation and application skills* by publishing products related to real-world situations (e.g., digital portfolios, games and simulations), and they *understand the impact of unethical use of digital tools*."

North Carolina Computer/Technology Skills Grade Level Competencies (Revised 2004): "The Computer/Technology Skills Course describes the *progressive development of knowledge and skills in six strands: Societal and Ethical Issues, Database, Spreadsheet Utilization, Word Processing/Desktop Publishing, Multimedia/Presentation, and Telecommunications/Internet*"

Example State Standards

California Career Technical Education (May 2005): *Information Technology Industry Sector* identifies 4 Career Pathways: *Information Support and Services, Media Support and Services, Network Communications, and Programming and Systems Development*.

Florida Teacher Certification Examinations (FTCE): standardized tests used to assess the competencies of prospective teachers. FTCE has 47 different exams: four General Knowledge, one Professional Education, and 42 Subject Area examinations. *Computer science is one of the 42 subject area examinations*.

Realities: Public Confusion

Six common public "definitions" of CS in the US:

1. CS = programming
2. CS = computer literacy
3. CS = a tool for studying science
4. CS = IT
5. CS is just for caucasian males
6. CS is not a science

Public Confusion Creates Bad Policy Decisions

August 26, 2009

"Recently the Kansas Board of Education … concluded that *the technology requirement is outdated* and that the content is being taught in other courses. Based on this conclusion, the Board is *proposing to cut the computing technology requirement*.

"It turns out that *while the technology requirement was intended to be a basic computing literacy course, it allowed many high schools to develop courses with computer science content*. ACM and CSTA's concern is that *if the Board eliminates the computing technology requirement* students will focus only on the core requirements and *K-12 computer science in Kansas will disappear*."

Cameron Wilson

ACM Director of Public Policy

Public Confusion Contributes to Equity Issues

K-12 CS is often not accessible to women or members of ethnic minorities.

Here's a 2008 example in a diverse Georgia high school (41% Caucasian, 59% African American and other):

- An AP CS course carrying science credit:
  - 33% Women
  - 10% African American and other
- A parallel Business Essentials course (no science credit):
  - 75% African American

Realities: Elective vs Core Subject

Scientists and engineers often intermix the terms "technology literacy" and "computer science" as if the two were the same.

- At NSF, *STEM* means "Science, Technology, Engineering, and Mathematics."
  - The place of CS in STEM isn't always clear.
  - CS can be viewed as a supporting skill for STEM.
- For K-12, this view translates to the notion that CS is an "elective" rather than a "core" subject.
- When offered as an elective, CS is difficult for K-12 students to fit into their already-crowded schedules.

Realities: "No Child Left Behind"

Adopted by Congress in 2002, NCLB requires states and school districts to:

- use standardized tests for all students in Math, Reading and Science (excluding CS)
- make students "technology literate" by 8th grade
- measure schools' "adequate progress" in Math and Reading

Impact of NCLB on CS:

1. Focus on core courses and standardized testing starves electives like CS for resources.
2. Focus on technology minimizes understanding of CS.
3. Eliminates any mandate to train *bona fide* CS teachers.

Realities: the AP Conundrum

Enrollments in the Advanced Placement (AP) computer science exams dropped by 15% during the period 2002-2008.

- Particularly discouraging is the low participation rate by females (17%) and ethnic minorities (11%).

In 2008, the College Board cancelled the Advanced AP Computer Science Exam (The regular exam is still administered).

- The content of that exam is limited by the narrowness of the introductory CS course offered at most universities.
- This contributes to the many public misunderstandings about the nature and importance of CS in the world.

Realities: Teacher Training and Certification

With no state curriculum standards, teacher preparation in CS in schools of education is nearly non-existent.

- A CS major in college can earn a far higher salary practicing CS than teaching CS in high school.
- Any certified high school teacher can offer a CS elective.
  - AP CS courses are usually staffed by math, science, and vocational education teachers.
- Most teachers and administrators don't understand the CS requirements in their own states (such as they exist).

Realities: CS Teacher Isolation

Excerpts from an e-mail written by a high school teacher in Maine (August 2009):

"Currently we require only a single quarter credit (9 weeks) of instruction for graduation, and this during the freshman year. *This is woefully inadequate.*

I currently teach three CS courses at the high school level.

I am trying to move our program from a purely elective program to one which requires credit for graduation in computer science.

I see education at the 9-12 level as being central in promoting more study at the college level.

Do you know of any other schools in the state or even the country at the 9-12 level that are working on a similar set of goals?"

Challenges and Opportunities

Despite these realities, many **grass-roots efforts** have begun to help change the landscape for K-12 CS.

- Many of these efforts have been inspired by CSTA initiatives.
- These efforts are beginning to turn the national tide for K-12 CS.

These efforts were recently joined by three **new national initiatives**:

- Computer Science Education Week
- A new NSF teacher development program
- A new AP CS curriculum

Opportunity: Teacher Development Institutes

Several initiatives have been taken by universities to help provide in-service training for K-12 computer science teachers.

- Georgia Tech is working to improve the quality and quantity of AP CS teachers, as well as increase the number and diversity of computer science students in Georgia.
- The University of Texas at Austin runs workshops for high school teachers. Its 2010 workshop will help teachers to:
  - Learn about new technologies that impact student learning
  - Learn about trends in CS education at the university level
  - Learn about the CS job market
  - Exchange experiences and teaching methods
  - Build relationships with UT-Austin faculty

Opportunity: On-Line Communities and Teaching Materials

The Computer Science Unplugged program (csunplugged.org) supports the group http://groups.google.com/group/cs-unplugged-sharing to encourage the sharing of CS teaching materials among K-12 teachers.

The CMU CS4HS web site http://www.cs.cmu.edu/cs4hs supports on-line K-12 teacher communities' sharing of computer science teaching materials and experiences.

The CSTA Source Web Repository is a searchable database with hundreds of unique resources: lesson plans, modules, and presentations for K-12 teachers. For more information, see http://www.csta.acm.org/WebRepository/WebRepository.html

Opportunity: Humanitarian FOSS

Free and open source software development serving humanity (H-FOSS) is a new approach to teaching software development in post-secondary CS programs.

- Many successes (e.g., Sahana, OpenMRS, Collabit)
- Goal: to help revitalize and diversify undergraduate computing education by adding a service learning component
- Supported by NSF CPATH, Google, Red Hat, Mozilla, etc.
- *Principles of H-FOSS are applicable to high school CS curricula as well.*

See (4) and hfoss.org for more information.

Opportunity: Pair Programming

An essential element of software development is the idea that students contributing to large open source software projects must work in teams and not in isolation.

One instructor puts it this way:

> "T*he benefits of higher confidence levels and more concerted efforts to develop a solution before turning for help vastly outweigh the occasional inequities that occur within groups*.
>
> "On a selfish note, it also means half the number of labs to grade!"

Opportunity: Recently, the CSTA has:

- Conducted 90 workshops to help teachers improve knowledge and skills in teaching CS
- Distributed careers brochures and posters to every secondary school in the US
- Created a cohort of master teachers who are working for improvements at the local level.
- Published three support documents for teachers wishing to use the Model Curriculum.
- Worked with the College Board and the NSF to create a new AP CS course that will be rigorous and engaging for all students.

Opportunity: CS Education Week

The US Congress designated the week of December 6-12, 2009 as "Computer Science Education Week."

It recognizes the transformative role of computing and the need to bolster CS at all educational levels.

Recommendations for policy makers:

Make CS a core course in the high school curriculum, introduce computational thinking in K-8, expand CS teacher development, ...

Recommendations for teachers:

Create better community awareness, introduce more CS activities for students, review cs videos from UW, CMU, ...

For more details, see the site csedweek.org, which is sponsored by the ACM, CSTA, CRA, NSF, Google, Intel, and Microsoft.

Opportunity: A New NSF Program

The new "CS/10K" program program promises to "catalyze a clean-slate revamp of high school computing education."

Goal: *to develop an effective high school curriculum that will be taught by 10,000 well-prepared teachers in 10,000 high schools by the year 2015.*

The new curriculum will feature two courses:

1. *an introductory course* – focus on computational thinking. For both college-bound and career technical education students.
2. *a completely revamped AP course – non-programming-centric*. The existing AP CS A course will be a follow-on to this one.

Greatest challenge: scaling teacher preparation programs to reach 10,000 teachers by 2015.

The Main CS/10K Challenge

In-service and pre-service teacher preparation must:

- Pair face-to-face training with online support that includes curricula, materials, and social networking
- Establish professional development programs
- Build a public-private partnership for funding
- Use university faculty, undergraduate, and graduate CS students
- Enlist K-12 teachers who are interested in this new initiative.

For more information on the NSF CS/10K initiative, contact jcuny@nsf.gov

Opportunity: AP CS Curriculum

The new AP Course will focus on underlying principles of computation, including problem-solving, abstraction, algorithms, data and knowledge creation, and programming.

It will also explore the limits of computation, the breadth of applications, and related societal issues. It will be "relevant, inspiring, and rigorous."

It will be flexible, allowing teachers to define the scope and sequence of a class with a range of topics and activities.

Teachers will have access to a curricular framework, instructional materials, and a set of exemplars.

- One exemplar is the new "Exploring Computer Science" curriculum.

Opportunity: New Teaching Materials for the Model Curriculum (3)

"Exploring Computer Science" – *a course to develop the CS skills of algorithm development, problem solving and programming in a large and diverse high school community*.

- Developed via an NSF grant to the University of Oregon and the Los Angeles Unified School District.
- Based on the *ACM Model Curriculum Level III course*
- Contains lesson plans, activities, and rubrics for 6 areas:
  - Human-computer interaction, problem solving, web design, programming, robotics, and applications.
- The University of California will grant college admissions elective credit for students taking this course

See http://csta.acm.org/Curriculum/sub/ExploringCS.html

Conclusions

In 2003, K-12 computer science was not well-understood or valued in the United States.

The ACM K-12 Model Curriculum was designed to help change this situation. Since 2003, change has been gradual.

- *Big surprise*: most advances have been accomplished through grass-roots efforts, almost one school at a time.
- Top-down systemic buy-in for CS at the state level *has not yet occurred* in any significant way.
- The CSTA has become a major force for systemic change in K-12 computer science education in the US.
- Continued progress will depend highly on the success of new NSF and AP initiatives.

References

1. ACM Task Force: A Model Curriculum for K-12 Computer Science, http://csta.acm.org/Curriculum/sub/ACMK12CSModel.html
2. Bureau of Labor Statistics: Occupational Outlook Handbook, 2008-2009 Edition, http://www.bls.gov/oco/ocos267.htm
3. Goode, J.: Reprogramming College Preparatory Computer Science, Communications of the ACM 51, 11, 31--33 (2008)
4. Salus, P.: The Daemon, the Gnu, and the Penguin: How free and open source software is changing the world, http://www.groklaw.net/staticpages/index.php?page=20051013231901859
5. CSTA National Secondary Computer Science Survey: Comparison of 2005, 2007, and 2009 Results, CSTA (2009), http://csta.acm.org/Research/sub/CSTAResearch.html
6. Stephenson, C.: The State of K-12 Computer Science Education, http://csta.acm.org/Advocacy_Outreach/sub/CSTAPresentations.html