Computational Thinking (CT): On Weaving It In

Paul Curzon
Queen Mary University of London
Mile End, London, E1 4NS
UK
+44 (20) 7882 5212
pc@dcs.qmul.ac.uk

Joan Peckham
Harriet Taylor
NSF
4201 Wilson Blvd.
Arlington, VA 22230, USA
(703) 292-7344
jpeckham/htaylor@nsf.gov

Amber Settle
School of Computing
DePaul University
243 S. Wabash Avenue
Chicago, IL 60604, USA
(312) 362-5324
asettle@cdm.depaul.edu

Eric Roberts
Computer Science
Stanford University
202 Gates
Stanford, CA 94305-901
650-723-3642
eroberts@cs.stanford.edu

Categories and Subject Descriptors
K3.2 [Computing Milieux]: Computers and Education - *Computer and Information Science Education*

General Terms
Design, Experimentation.

Keywords
Computational Thinking, Curriculum, Education.

1. Summary

Computing offers essential problem-solving tools needed for contemporary challenges. The role of computing in education, and appropriate pathways for modern students, are of concern [10]. Educators recognize the importance of improving information technology (IT) skills and fluency, and studies have developed guidelines [7][8], but the analytical concepts and tools of computing have benefits beyond IT fluency.

CT [12] continues earlier discussions on the nature of computing, [3][9]. This has helped the computing community to strengthen definition of the problem solving skills that computing brings to society, through education, outreach, and research.

Recently, CT has served as a basis for several efforts aimed at more precise, deeper and wider interpretation of computing. This includes attention to K-12 curricula, general education at colleges and universities, and interdisciplinary research and tech. transfer. This panel discusses general US CT developments, then specific panelist activities, and ends with questions and discussion.

2. Paul Curzon

Paul Curzon and Peter McOwan have created the cs4fn magazine and approach [4][5], a great vehicle for implicitly communicating computing ideas to students. The philosophy behind cs4fn is that learning sinks deepest in hobby-mode; if people want to do it for fun, the learning will be easy. CT is a natural for this because people often do it for fun: Sudoku, cross references or pixel puzzles, magic tricks and chess strategies.

A special aspect of CT over other key skills is abstraction; finding the links between problems to reuse solutions. This idea runs deeply through cs4fn with often "wacky" connections used as the "grab" at the start of an article or talk: "What has Gray code to do with the Chinese rings puzzle?", "What has the game of 20 questions to do with helping someone with locked-in syndrome?", "What has the magic trick we just showed you to do with computer programs?", "What has Spanish flu to do with World of Warcraft?". Exploring the connections between apparently different problems not only teaches computing concepts but also implicitly reinforces the CT message. CT is further enforced in our live shows that follow the Computer Science Unplugged [1] approach, using kinesthetic activities and physical analogues [6] and even magic tricks [5] to illustrate ideas. This shows the mapping between problem domains that is core to computer science and illustrates abstraction in a physical and natural way.

We explicitly present computing as inter-disciplinary; each issue of the magazine looks at the overlaps between computing and another subject. The beauty of computing that it is engineering, science, and art; a subject without clear borders and with tentacles pushing into every other subject. It is CT, the different way of thinking about problems and their solutions, that allows computing to transform whatever it touches. This interdisciplinary approach gives us the opportunity to reach students with interests outside computing and introduce them to computing and CT.

Creativity is an important part of the CT skill set. By showing how real CT-ers (industrialists, professors, male and female, all ethnic backgrounds) have found creative solutions to real-world problems, breaks stereotypes and attracts females.

If the CT agenda is to be used in a positive way to reignite interest in the subject, we need to find creative ways to sell it, to enthuse students, and then to teach them the skills and imbue them with the computing philosophy when they do chose the subject. The cs4fn approach seems to be one good way to do that.

3. Amber Settle

Faculty in the School of Computing and the School of Cinema and Interactive Media in the College of Computing and Digital

ITiCSEe'09,July 3-8, 2009, Paris, France.

Media (CDM) began a project in August 2008 to integrate CT in various areas of DePaul University's Liberal Studies Program. Over the past 5 years, CDM faculty have developed nearly 50 liberal arts courses. These courses are offered in nearly every area of the Liberal Studies Program, including Arts and Literature, Understanding the Past, and Philosophical Inquiry, and were developed in partnership with many academic units outside of CDM. Primarily developed for liberal arts concepts and skills, these and other courses developed by CDM include CT elements.

In the first year of the project, CDM faculty will work to make CT explicit in a selection of the courses, to develop tools to evaluate the learning of CT, and to create a framework that can be used by non-technical instructors to integrate CT in their courses. The courses and participants from within CDM are chosen to facilitate transition to non-technical courses. CT will be enhanced and expanded in 10 CDM courses in areas including computer science, IT, e-commerce technology, animation, computer game development, and digital cinema. Dr. Settle will highlight CT activities, assessments, and learning goals from these courses and discuss the initial framework and data gathered in Spring 2009.

Faculty from outside the College of Computing and Digital Media and from outside DePaul University will participate in the project during the second year. We would particularly like to see the integration of CT in the humanities, arts, and social sciences. DePaul faculty from anthropology and history have already indicated interest in the project, and we have identified partners at the City Colleges of Chicago and Illinois Institute of Technology

4.0 Harriet Taylor

The CSTB (Computer Science and Telecommunications Board) in the Division on Engineering and Physical Sciences in the National Academy of Science is hosting an NSF funded workshop series, (http://www8.nationalacademies.org/cp/projectview.aspx?key=48969), CT for Everyone in 2009. Two workshops will explore the nature of CT and its cognitive and educational implications. Workshop 1 focused on the nature of CT, inviting computer scientists to ponder what "CT for everyone" means. It also involved a few cognitive scientists, and educators to ground the discussion in the cognitive realities of K-12 children. Workshop 2 will focus on the cognitive and educational dimensions of CT, will invite scientists to explore the intellectual readiness to handle different aspects of CT at different ages with different backgrounds and consider how CT might be incorporated in K-12 education. For each workshop, a report summarizing discussions, contrasting the different views, and identifying commonalities will be prepared by the workshop committee and widely disseminated in the computer science and education communities to stimulate further discussion about CT. Dr Taylor will give a preliminary report of the first meeting.

5.0 Eric Roberts

For at least 30 years, the CS department at Stanford has encouraged students into their introductory courses to implement "computational thinking everywhere". There have been ebbs and flows in enrollments, but more than 70% of Stanford undergraduates in the last 20 years have taken at least one CS course, even though not required. These courses are designed to appeal to a broad audience -- to be "funnels" rather than "filters."

Stanford has deliberately chosen not to offer a CT course for non-majors to avoid the following pitfalls: 1) students using computers and who are insulated from programming are likely to be pushed away from a computer science major just when the economy needs more computer scientists 2) not teaching programming sends the message that programming is as unpleasant as its current reputation. Our courses emphasize the "passion, beauty, joy, and awe" that is as much a part of programming as it is of the rest of the field. [2]

Although CT does not mean that all students must master the details of a complex programming language, it is essential for students to understand the ideas of programming, to appreciate the difficulty of expressing a solution in a form that can be interpreted by a machine, and to have some experience with debugging [11]. The danger in courses that divorce CT from programming is that students come to see programming as low-level and mundane, rather than as creative and intellectually exciting.

Making CS classes for everyone automatically increases the diversity of our courses and our major. These courses are more likely to reflect the composition of the whole student body than if they are designed to appeal to a small elite.

6.0 ACKNOWLEDGMENTS

A. Settle thanks NSF CPATH for #0829671. cs4fn is supported by EPSRC #EP/F032641/1, Google, ARM, and Microsoft.

7.0 REFERENCES

[1] Bell, T., Fellows, M. and Witten, I. 1998. Computer Science Unplugged: Off-line Activities and Games for All Ages. Available at: www.lulu.com

[2] Booch, G. 2007. Read'n, writ'n, 'rithmetic, and cod'n. Keynote address at the Thirty-Eighth SIGCSE Technical Symposium on Computer Science, Houston, TX.

[3] Bundy, A. 2007. Computational Thinking is Pervasive. Journal of Scientific and Practical Computing. 1,2 67-69.

[4] Curzon, P. Serious Fun in Computer Science, ACM SIGCSE Bulletin 39(3) p1. ITiCSE07 Invited Keynote.

[5] Curzon, P. and McOwan P.W. 2008. Engaging with Computer Science through Magic Shows", inroads SIGCSE Bulletin, 40 (3), 179-183. ACM

[6] Curzon, P. and McOwan, P.W., Cutts, Q. and Bell, T. 2009. Enthusing & inspiring with reusable kinaesthetic activities, In proceedings of ITiCSE09.

[7] CSTB, National Research Council. 1999. Being Fluent in Information Technology. National Academy Press.

[8] Computing Curricula 2001: Computer Science. IEEE Computer Society and the ACM.

[9] Denning, P. 2005. Is Computer Science Science? CACM. 48,4 (April 2005). 27-31.

[10] Guzdial, M. Paving the Way for Computational Thinking. CACM. 51,8 (August 2008), 25-27.

[11] Nat'l Res. Council. Being fluent with information technology. Washington, DC: Nat'l Academy Press, 1999.

[12] Wing, J. 2006. Computational Thinking. CACM. 15, 5 (March 2006), 33-35.