Computing for Everyone: Improving Global Competitiveness and Understanding of the World

A White Paper for ICER Workshop

Mark Guzdial, Georgia Institute of Technology

The goal of computing education for the next five to ten years is to ***establish computing as part of a liberal, general education***. Like mathematics or laboratory sciences, taking computing courses should be a presumption of an educated professional or academic. Everyone would take some computing, and most would take more.

There are three reasons for the necessity of this goal. The first is that that's where the future jobs are, in the mix of computing with other disciplines. As Thomas Friedman argues in his book *The World is Flat* (Friedman, 2005), forces of global competitiveness require future workers to become *Versatilists*. "The world 'versatilist' was coined by Gartner Inc., the technology consultants, to describe the trend in the information technology world away from specialization and toward employees who are more adaptable and versatile…Enterprises that focus on technical aptitude alone will fail to align workforce performance with business value." Even technological powerhouses like Microsoft is looking for versatilists, as Bill Gates said, "The nature of these jobs is not closing the door and coding. The great missing skill is somebody who's good at understanding engineering and bridges that to working with customers and marketing (Montalbano, 2005)."

The second reason is that a liberal education is about understanding one's world, and computing is a huge part of today's world. We ask students to take laboratory sciences (like biology, chemistry, and physics) in order to better understand their world and to learn the scientific method for learning more about their world. The virtual world is an enormous part of the daily lives of today's professionals. Understanding computing is at least as important to today's students as understanding photosynthesis.

The third and most significant reason is to meet the potential of computing and other fields. Alan Perlis first called for computing as part of a liberal education in 1961 (Greenberger, 1962). He argued that the ability to specify and execute process would offer a whole new method of exploring domains and learning. If you understand something well, you should be able to define its process well enough for a machine to execute it. If you can't, or the execution doesn't match the observed behavior, we have a new kind of feedback on our theories. He used examples from economics in his 1961 talk, but the entire field of computational science demonstrates his prescience. Computing has enormous potential in many fields where its use today is limited to whatever Microsoft Office can do.

Computers are workhorses for plowing mental fields, but to harness these beasts of burden, one has to know how to command them. Most professionals today are limited to using applications developed by technology specialists, who can't possibly understand all other professions as well as their practitioners. To follow the analogy, it's as if farmers and mill owners of the past were told, "Look, horses are only good for pulling wagons for merchants. I can't help you with your plows and mills, but I can sell you wagon." The potential for computing in our society will only be met when all professionals have the capacity for understanding and commanding computing workhorses. When educated people across our society realize what computing really can do for them, the demand for software development professionals will increase to scale and disseminate the good ideas of the versatilists.

Our current computing education cannot meet this goal. Our track record for educating students about computing is dismal. We can't attract and retain the students who claim that they want to focus on computing, and it's much worse with non-majors. The percentage of women and minorities taking computing courses continues to drop, even with all the attention paid to it.

The largest change that must occur in our computing education in order to create computing for everyone is to recognize that the goal of computing education is not *only* to produce software development professionals. Creating software development professionals will be a fraction of the challenge of our education task if everyone on every campus studied computing. We overemphasize techniques and methods for large scale software development in our classes, which are not the most important benefits that we have to offer the rest of academia.

A computing for everyone should emphasize the laws, limits, uses, and wonders of computing. A few examples include:

- That we can define better or worse processes, and that processes can be *proven* correct (something that Perlis thought everyone should be taught).
- That there are processes that can't be successfully defined like a solution to the Halting Problem, or if defined for a computer to execute, may not finish in your lifetime.
- That the line between 'program' and 'data' is permeable, and that exploiting that permeable boundary is how many viruses attack.
- That information, once digitized, can be mapped and re-encoded into other media, forms, and representations.

Our proof of concept is the new courses at Georgia Tech's College of Computing. We have been creating *contextualized computing education* where we teach non-CS majors in classes that draw on relevant examples and uses in their field and that emphasize computing concepts and skills that go beyond just software development. We teach engineering students MATLAB with engineering-oriented problems, and we teach management, architecture, and liberal arts students computing for creating and manipulating media (Guzdial, 2003). We are enjoying dramatically higher retention rates in these contextualized courses than in more traditional computing courses, with women and minorities succeeding at the same rates as white men (Rich *et al.*, 2004). It's a portable innovation: the courses are being adopted at other institutions with similar improvements in retention (Tew *et al.*, 2005). But most importantly, follow-up studies have students telling us a year later that the courses have changed how they think about computing and use it in their daily lives (Guzdial & Forte, 2005). GT's new *BS in Computational Media* degree, a versatilist combination of computing and liberal arts, drew over 100 majors in the first year and is nearly one-quarter female. Our field needs similar innovations in many contexts to draw in everyone across the academy into computing.

References

Friedman, T. L. (2005). *The world is flat: A brief history of the twenty-first century*: Farrar, Straus, and Giroux.

Greenberger, M. (1962). *Computers and the world of the future*: MIT Press.

Guzdial, M. (2003). *A media computation course for non-majors.* Paper presented at the Proceedings of the Innovation and Technology in Computer Science Education (ITiCSE) 2003 Conference.

Guzdial, M., & Forte, A. (2005). Design process for a non-majors computing course, *Proceedings of the 36th SIGCSE technical symposium on Computer science education*. St. Louis, Missouri, USA: ACM Press.

Montalbano, E. (2005, July 18). Gates worried over decline in us computer scientists. *Computer World*.

Rich, L., Perry, H., & Guzdial, M. (2004). *A CS1 course designed to address interests of women.* Paper presented at the Proceedings of the ACM SIGCSE Conference.

Tew, A. E., Fowler, C., & Guzdial, M. (2005). Tracking an innovation in introductory cs education from a research university to a two-year college, *Proceedings of the 36th SIGCSE technical symposium on Computer science education*. St. Louis, Missouri, USA: ACM Press.