Future Challenges in Computing Education

White Paper by

Jaime Seguel, PhD
Professor of Computer Science
ECE Department and
Director of CISE PhD Program
University of Puerto Rico at Mayagüez

Faced with a declining undergraduate enrollment Computer Science educators start wondering whether the youngster’s enthusiasm for computing disciplines in the eighties and nineties was indeed the sign of real trend or just another “computing fade”, much more like the ill-famous “dotcom bubble”. The fact is that Information Technology industry is still a vibrant economic force worldwide, and still an economic force that depends crucially on a mature and well-rounded human and intellectual capital. And universities are still main contributors to both of them. Thus, this decline in the youngster preference for Computer Science is unlikely to be motivated by a pessimistic view of the professional expectations but rather, by what they expect to acquire during their term in the university. We have to admit at this point, that there always has been a distinguishable gap between university studies and professional practice, so in this respect, Computer Science is not alone. However, in Computer Science education, because of its very nature of young-in-the-making discipline, this gap widens at a much more accelerated pace than in other Science and Engineering disciplines. The substantial difference between the IEEE and ACM Computing Curricula editions is a clear indicator of this phenomenon. And I believe that this is also, the ultimate reason behind the decline in undergraduate enrollment in Computer Science in USA. At this point in time, the gap seem to be wide enough for students to find little correlation between what are them taught and what they perceive as a valuable professional skill or intellectual goal. Although it may be rightfully argue that education is not just a vessel to be filled at instruction time with objects to be taken out of the vessel later, in the professional life but something that instills in the minds and spirits the essentials, the general concepts, understandings and ways of thinking of a discipline; still the question remains as to whether we are achieving that ultimate goal in Computer Science education. In what follows, I make a quick and by no means exhaustive reflection on two angles of this question.

First is the old tension between concretion and abstraction. Computer Science is born out of the convergence of a technology stream, the one that started with Babbage’s Analytical Machine and culminated with the ENIAC; and a mathematical stream that started with Hilbert’s formalist program and culminated with Alan Turing’s abstract machine. The richness of this convergence is seldom perceived by students. For most of them, there is no real connection between this two seemingly separated trends, and very much like most of their own
instructors, tend to declare themselves too early for their benefit, theoretical computer scientists, computer scientists, software engineers, or some other computing specialization. Everybody loses with an early super-specialization. A theoretically oriented computing specialist without a sound knowledge of hardware and software systems does not know the very object of her or his science. A computer systems designer without a fair knowledge of the theory of computing is unlikely to be able to cope properly with the overwhelming complexities and abstractions that are inherent to such systems. The advent of distributed computing raises exponentially the systems complexity adding a communications and time uncertainty component to the picture, and bringing about issues related to the design of self-healing, self-adjustment, self-managing machines. With all this, the need for a serious and well-educated abstraction in computer systems designers, far from being just an academic profile concern is turned into a very practical matter. In my opinion, failure to develop Computer Science curricula capable of teaching the richness and essentiality of this dual nature of the discipline is to a great extent, equivalent to losing the very essence of the Computer Science way of thinking.

Second is the fact the Computer Science is a hub for almost every technology and most modern science, business, industry, government, health and education. Computer Science is thus a universal enabler, the core of a whole movement, the information technology era. This is certainly the main reason behind the current importance of Computer Science in modern society and its rapid development and acceptance as an independent discipline. But being a universal enabler brings about some danger to the discipline, as well. Often, the enabled activity of discipline gets the credits, not the enabler. Furthermore, the enabled tend to absorb the enabler's products into its collection of tools and solutions. I remember the times when quite a few Computer Scientists devoted considerable effort to Computational Sciences problems. Today, most products of this research have been absorbed in the form of high-performance software solutions, and are not longer considered to be a central activity in Computer Science. Just as another example, I am currently involved in the public launching of a Grid Computing infrastructure. Although this infrastructure will be a test bed for several exciting problems in distributed computing we were advised not to emphasize this in the launching but what the infrastructure could do for solving hazard mitigation, health care and bioinformatics problems. That message certainly influences the decision of High School students intending to go to College, as well. I do not question the contribution of Computer Science to other disciplines and problems. What I believe is worth considering in future curricular developments is an expansion of traditional Computer Science, this is, the science and technology of information representation, storage and processing to the nature of information systems, both natural and artificial. Computer Science has developed the basic tools and concepts for such an expansion. I guess I mostly refer here to what is nowadays called informatics. I believe that embracing informatics, although maybe a bit of a departure from the traditionally established
Computer Science curriculum will renew the interest of High School students for undergraduate programs in Computer Science.